

Week15

1. Self-Attention

解决输入维度不同的内容 (eg 句子)

basic method: One-hot Encoding

Word- Embedded (给每个词汇一个向量, 距离短的语义近)

输出是每个输入对应的label (eg 词性标注 sequence labeling), 是一个label (情感识别), 模型自行决定多少个label (sequence2sequence)

Fully- connected: 全连接层相当于把不同的 feature map 打平展开为一个 vector, 然后用这个 vector 来做 classification

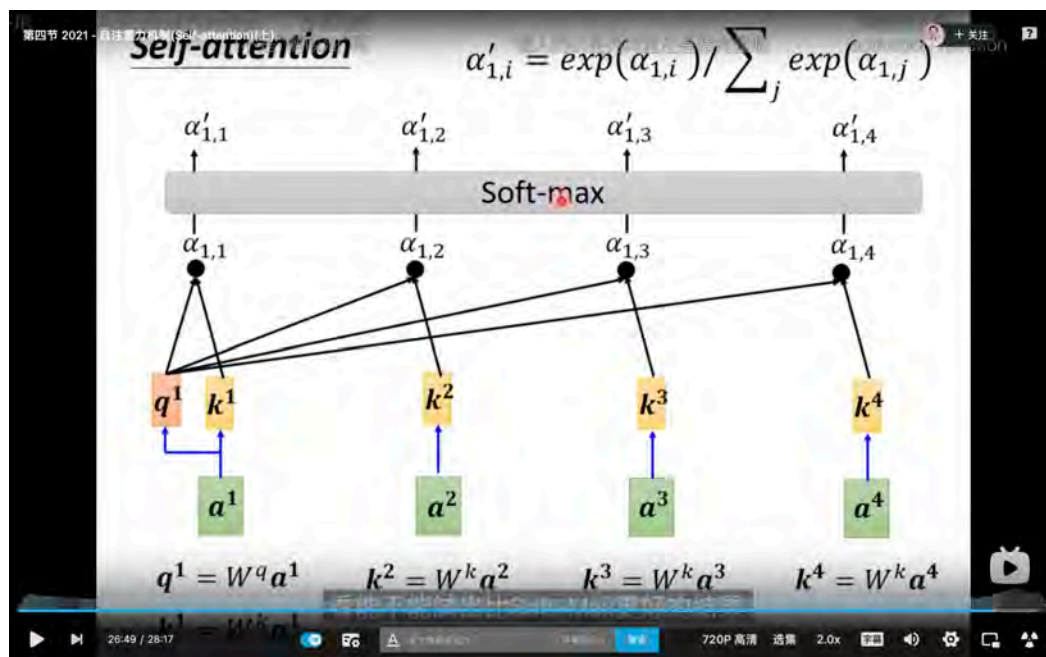
sequence labeling:

不能直接用 fully- connected layer 来面对每一个词定性 (丢失关联信息)

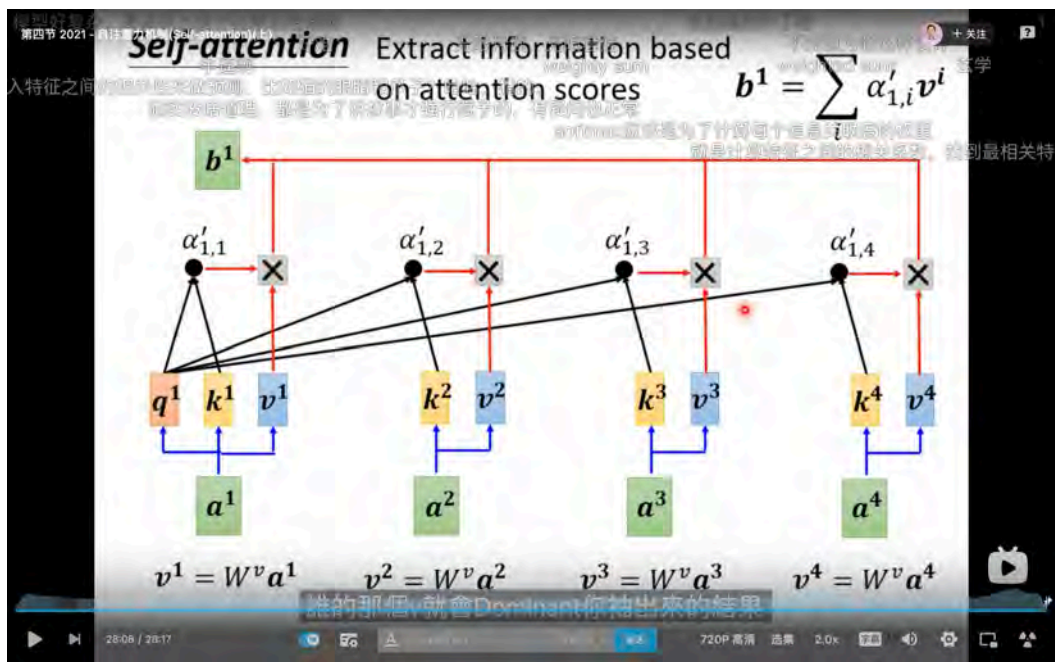
Self- attention 可以读取整个 sequence 的信息

用 α 来表示两个向量之间的关联性:

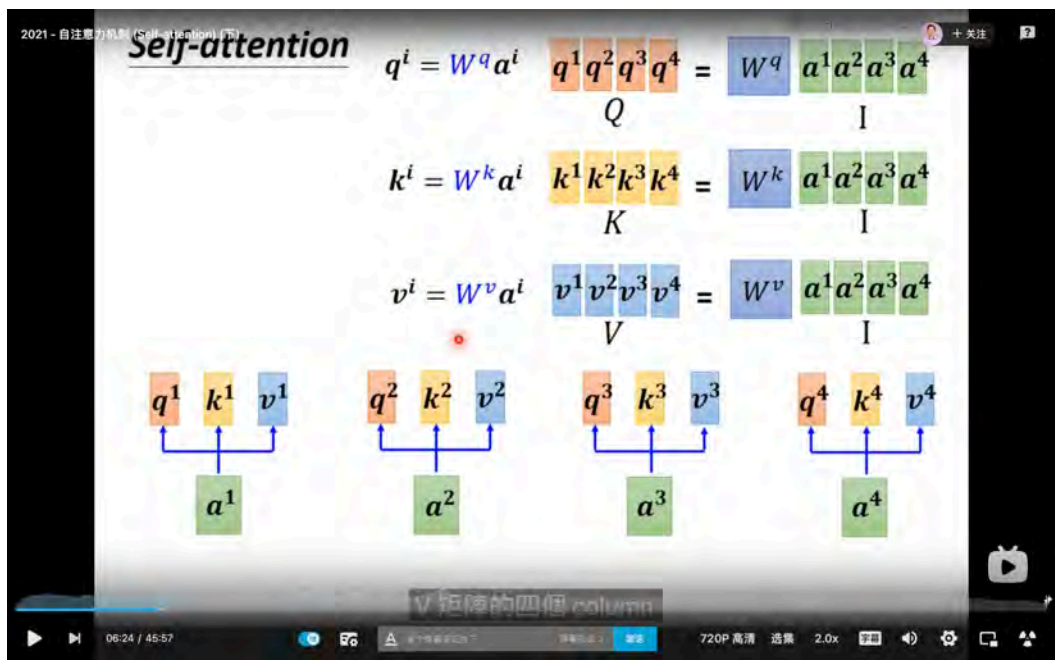
1. Dot product: 将求解关联性的两个向量分别乘上两个不同的矩阵 W , 再用结果做 dot product 得到 α (additive 也很常用)



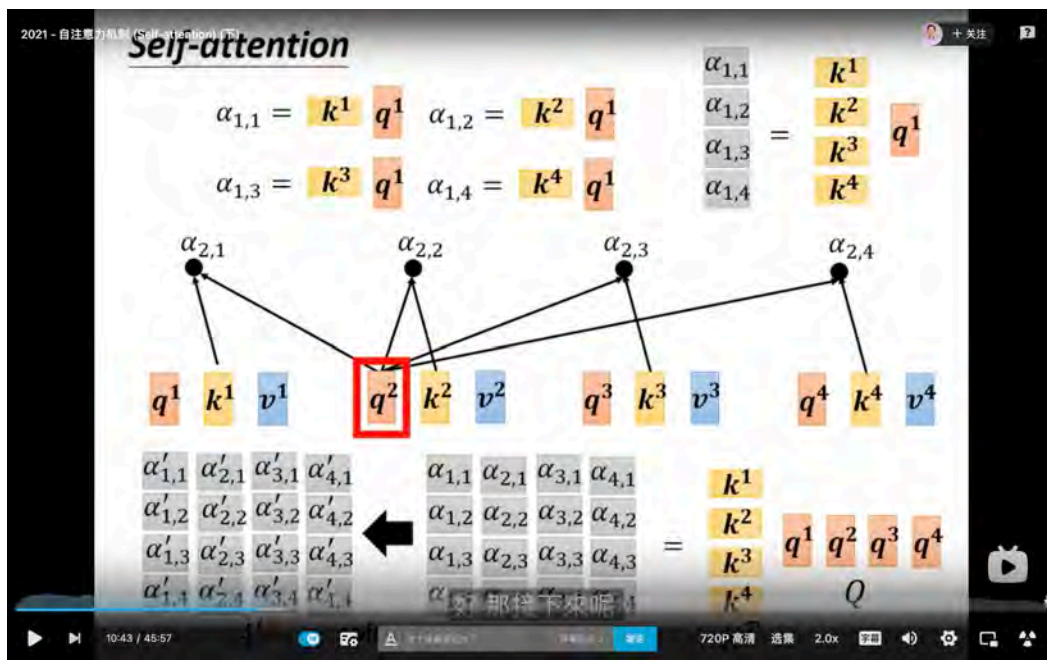
然后求加权和:



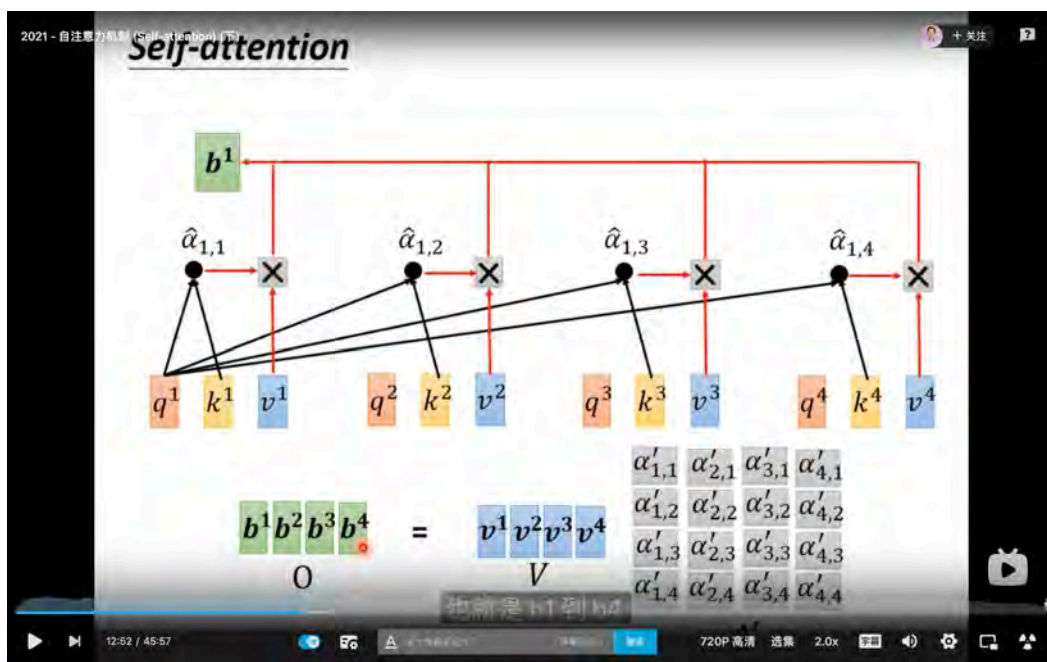
矩阵视角：



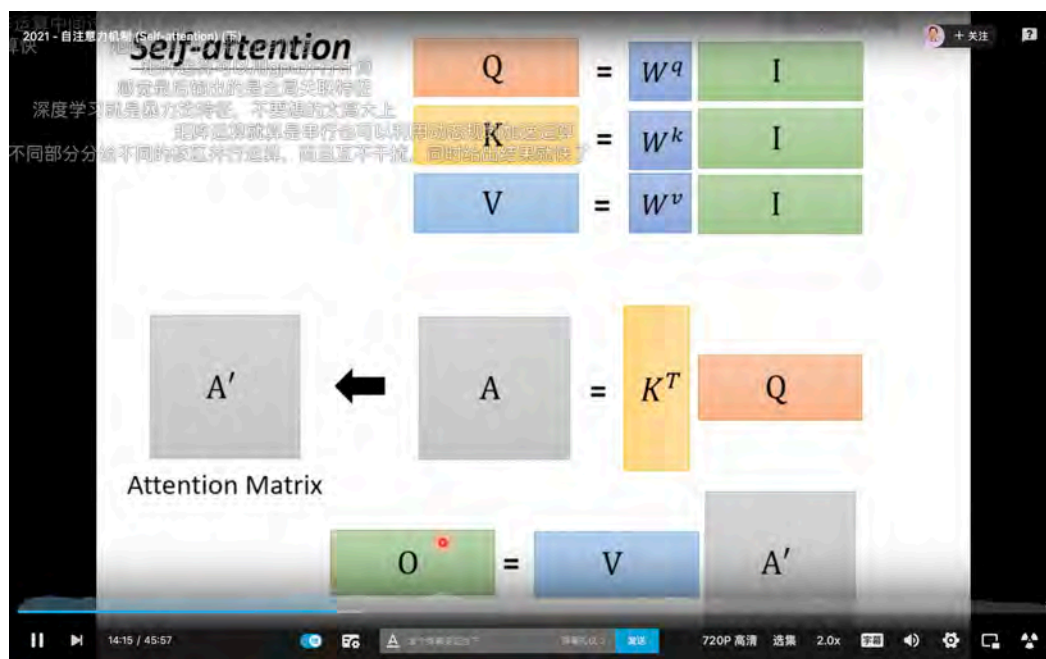
从 q k 到 注意力系数 α :



矩阵视角的b:



overall:



multi-head self-attention:

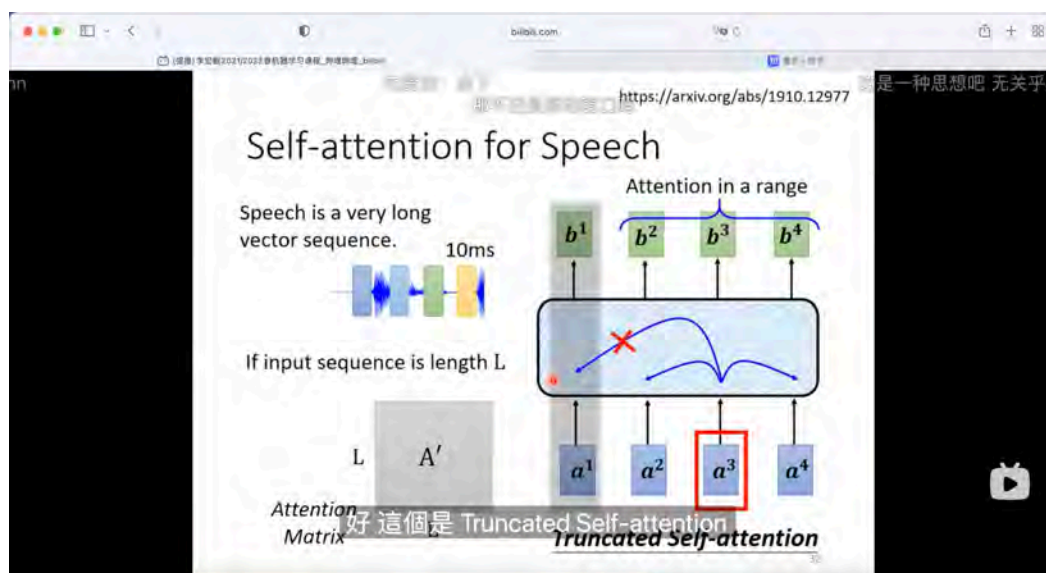
有多个 q 、 k 、 v ，可以同时考虑不同的相关性，一般要对结果 b_i 再用 W_o 矩阵变回特定维度（即 b_i 之间的加权平均）

Positional Encoding:

给每一个位置 encode 一个位置信息，加到 a_i 上（目前不够科学）

move to self-attention for speech:

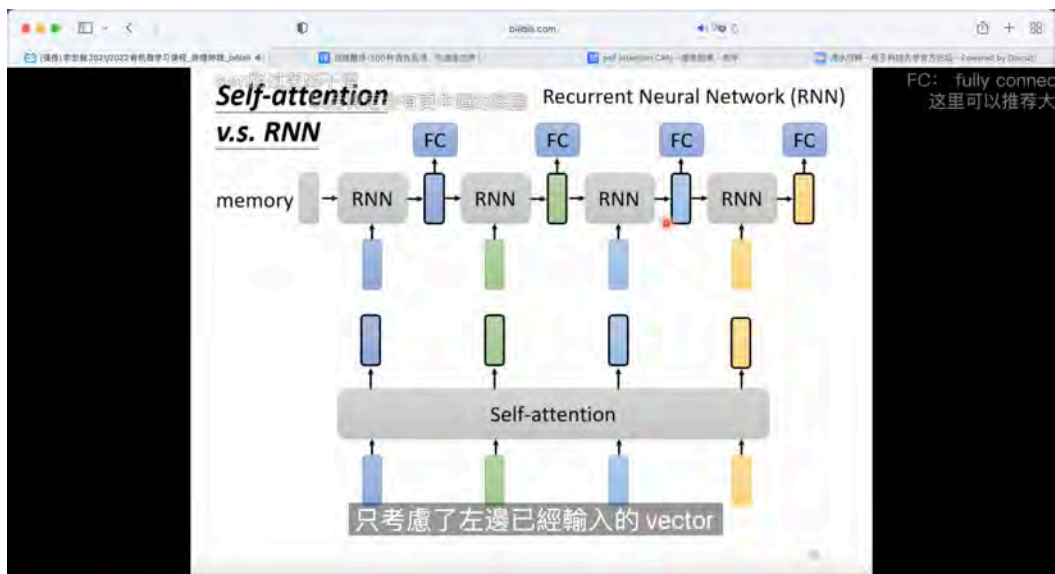
语音数据的 sequence 可能会非常的长，会导致我们的 self-attention 矩阵非常大，对内存要求高，所以我们可能需要人为设定一个范围



CNN vs self-attention:

CNN 是 self-attention 的特例

RNN vs self-attention:

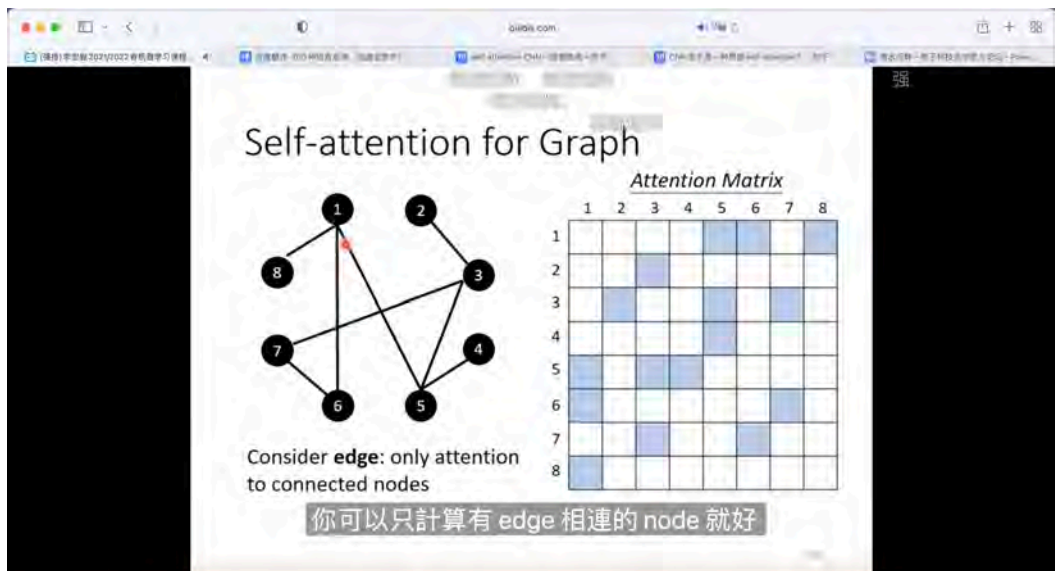


RNN不能并行计算，所以在深度学习时代被淘汰了

CNN和Self-attention不存在谁包含谁的问题，在某种设定下可以近似等价，这个要分情况来看。

1. CNN在使用静态卷积核 W 时，输入 X 是小局部方式、 W 是全局方式。 W 是一种参数，从整个数据中学习到的。
2. Self-attention则是输入 X 是大局部，注意力系数 W 大局部。
3. CNN在使用动态卷积核时，在某种设定下，可以近似self-attention。

graph neural network:



以上 self-attention

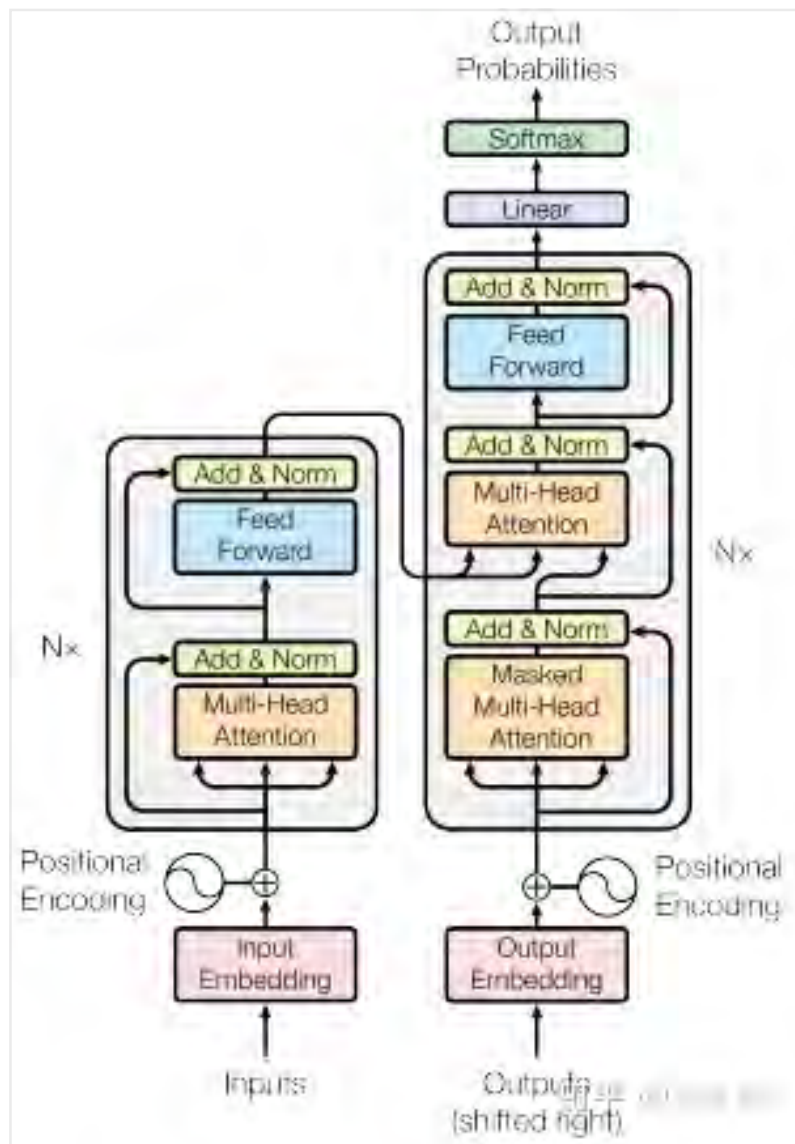
Transformer:

A Seq2Seq model: input a seq, output a seq

Application: translation, audio to words(Speech recognition), speech translation (硬train 一发), Text to speech, chat robot, QA, 文法剖析 (grammar as a

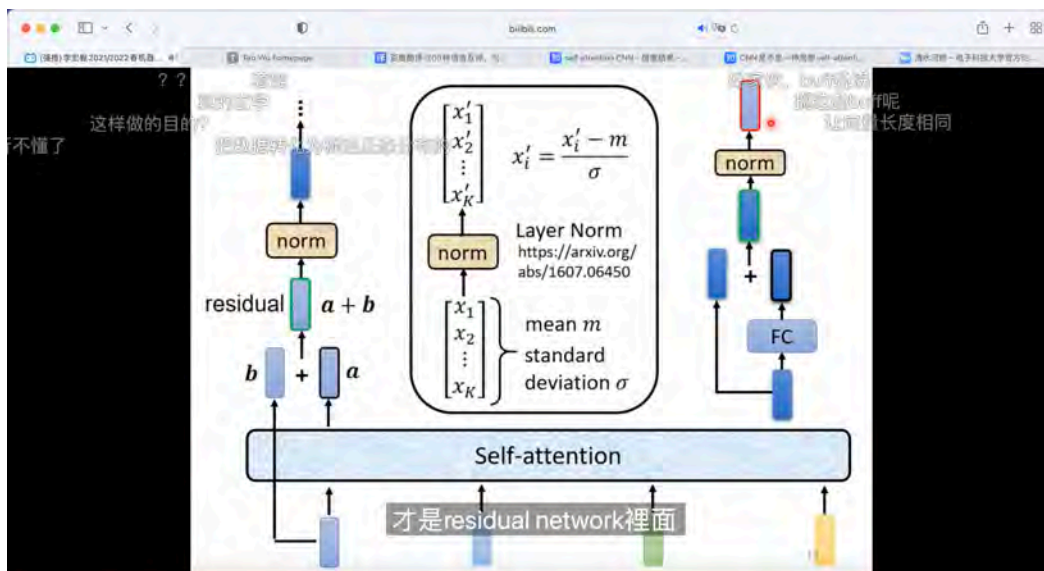
foreign language) , multi-label classification (多标签)

Transformer 的优势：可以并行 (RNN 不行)



架构：

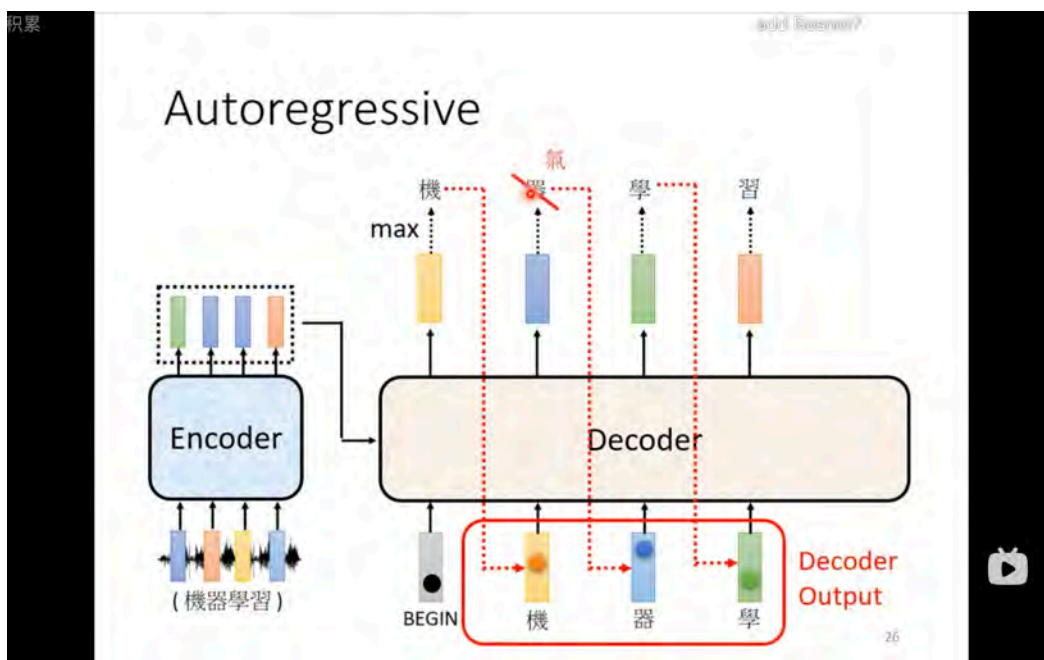
encoder: self-attention



输入一个 sequence 输出一个 sequence

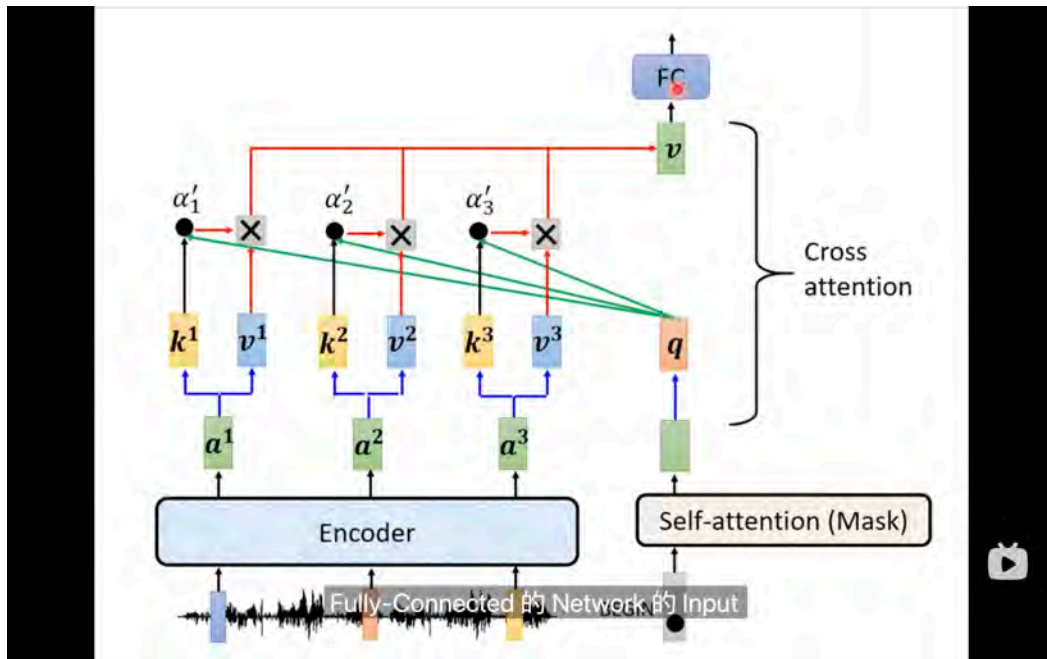
Decoder:

Auto-regressive (自回归)：在输出的时候我们只能一个一个词输出给 decoder 来作为输入，不能像 encoder 里面直接对所有进行 attention (所以要用 masked attention)



Decoder 决定 sequence 长度的方式：用一个和 begin 一样的全连接层输出 “Stop” 来终止序列

Not autoregressive transformer (NAT)：直接同步输入

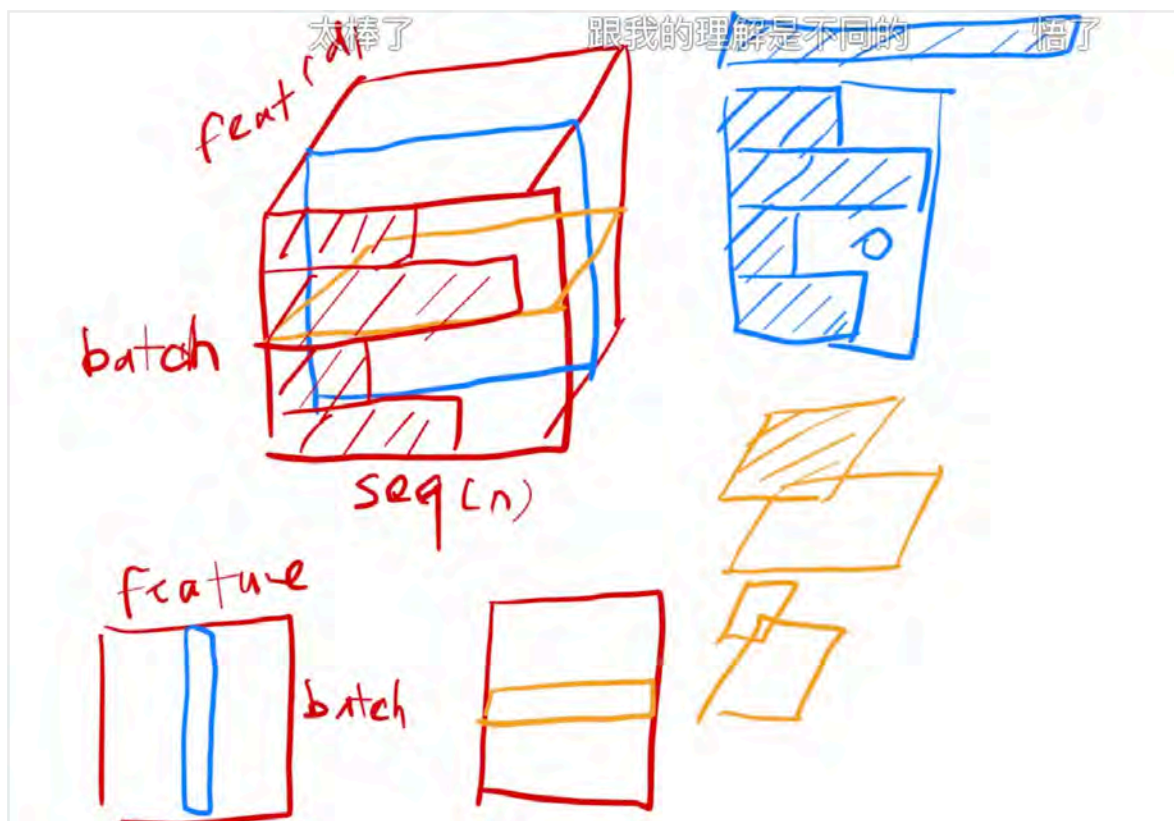


cross attention: decoder 怎么读到encoder的内容

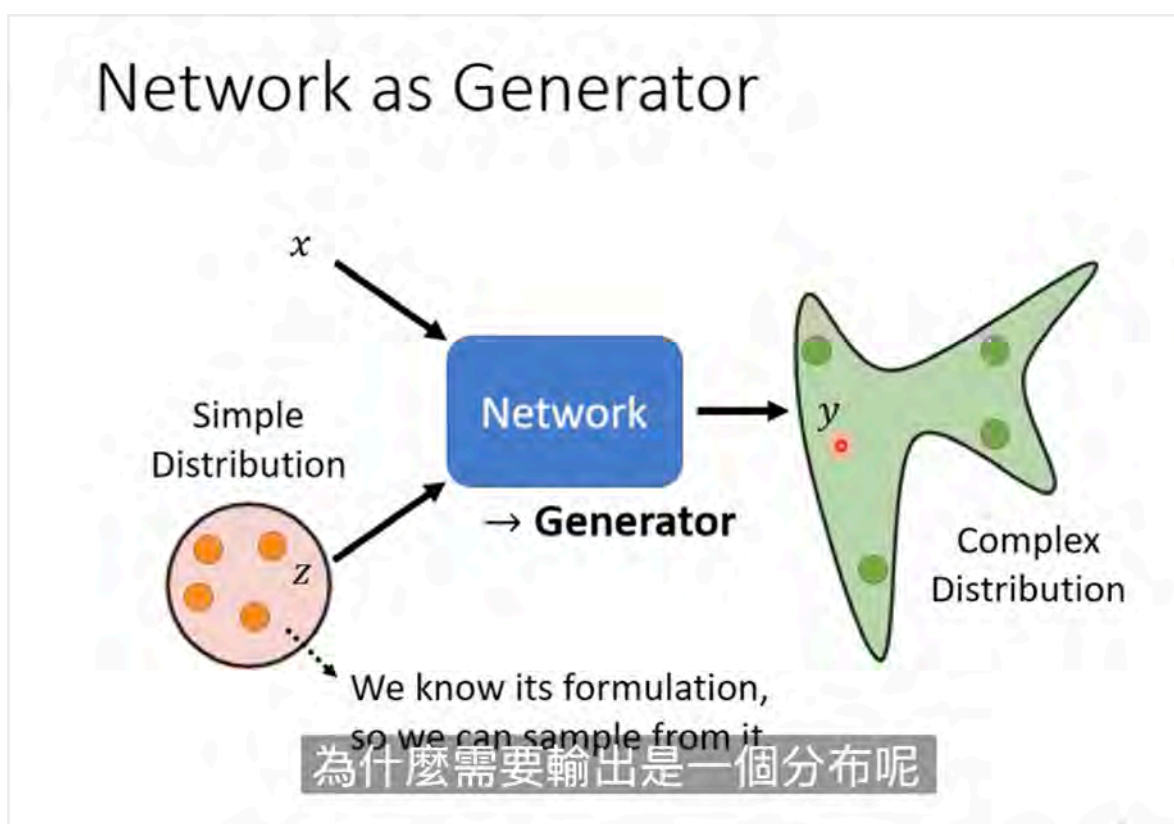
Copy mechanism

batch Norm vs layer Norm:

因为NLP一类的数据可能长短不一，所以用feature Norm的话可能对于长度不统一的量不准确，所以我们预测函数还是在batch内较好（BN解决的问题应该是不同通道的同类数据，比如图片的RGB，所以batch内做norm结果应该是make sense的）：

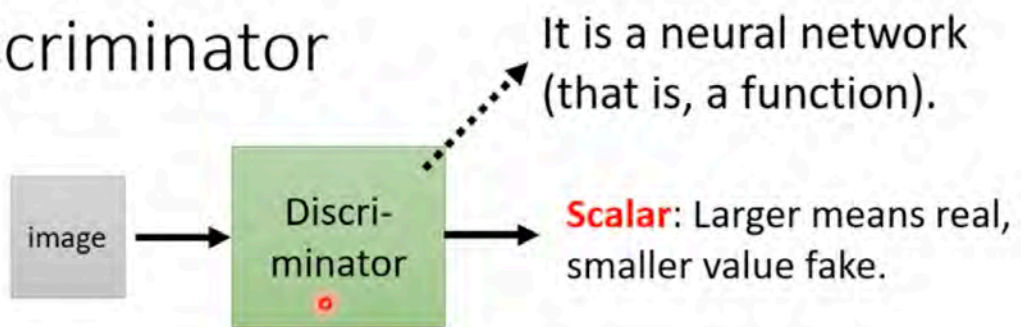


3. 生成对抗神经网络 (generative adversarial network, GAN)



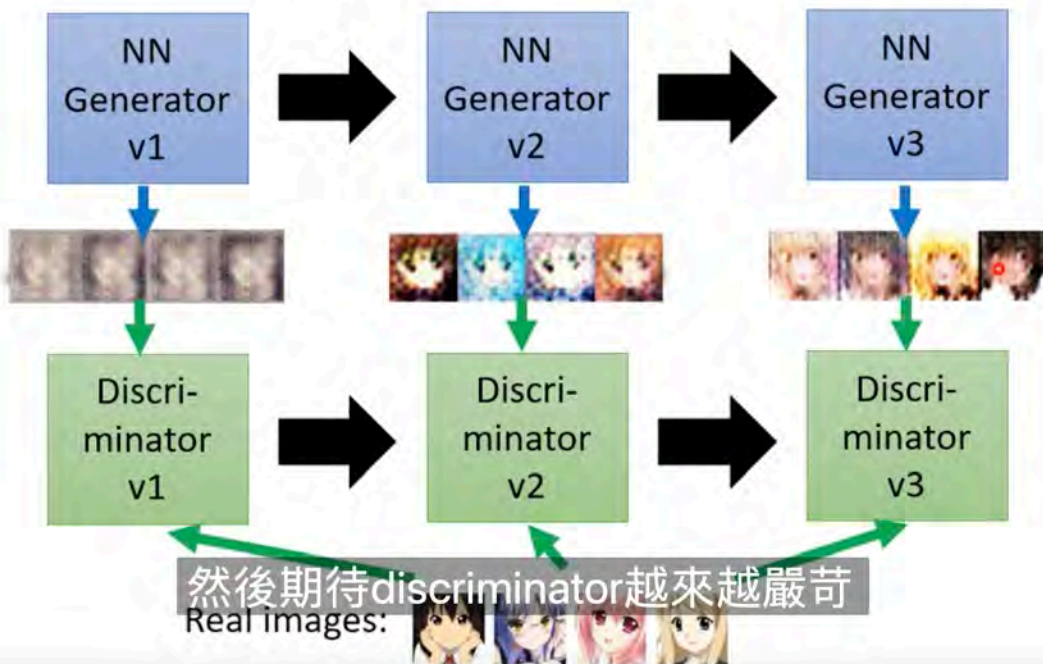
discriminator

Discriminator





合作：

Basic Idea of GAN

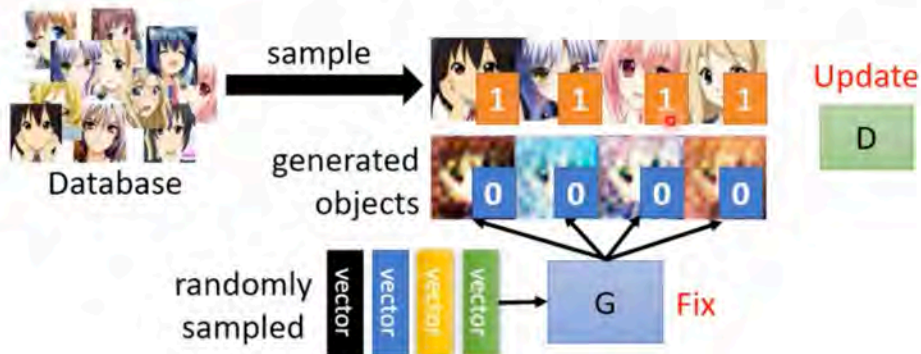


伪代码

Algorithm



- Initialize generator and discriminator  
- In each training iteration:

Step 1: Fix generator G, and update discriminator D



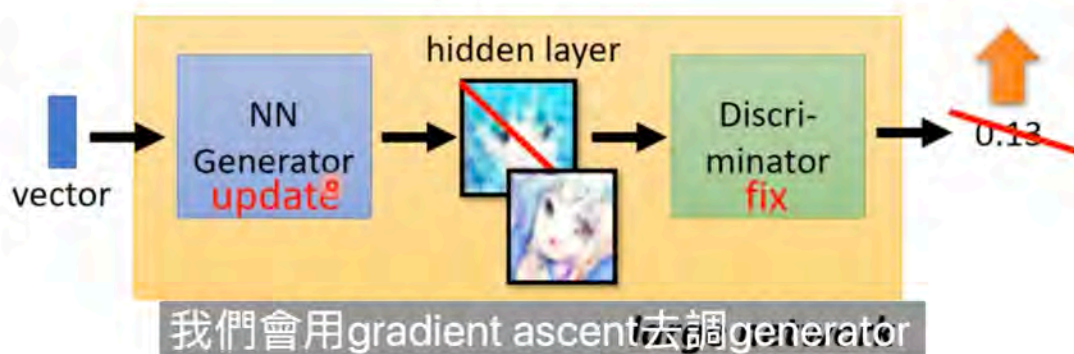
Discriminator learns to assign high scores to real objects and low scores to generated objects.

15

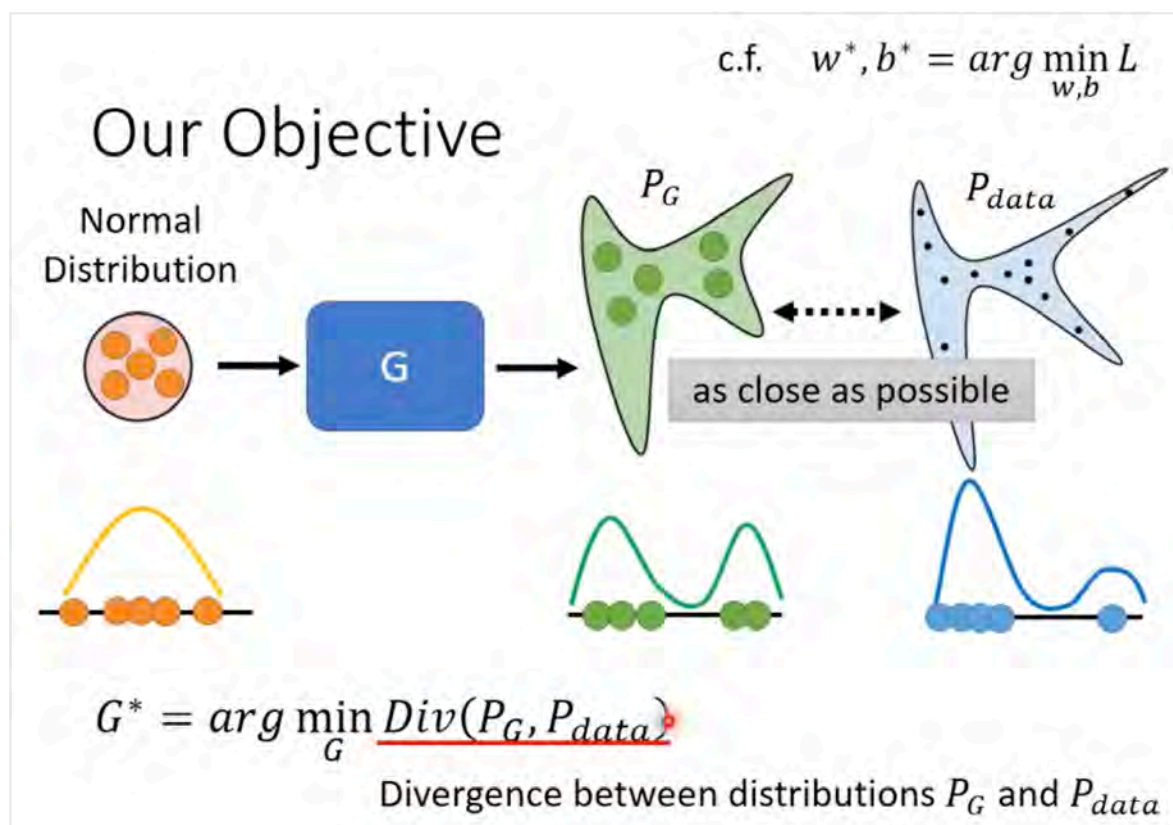
- Initialize generator and discriminator  
- In each training iteration:

Step 2: Fix discriminator D, and update generator G

Generator learns to “fool” the discriminator

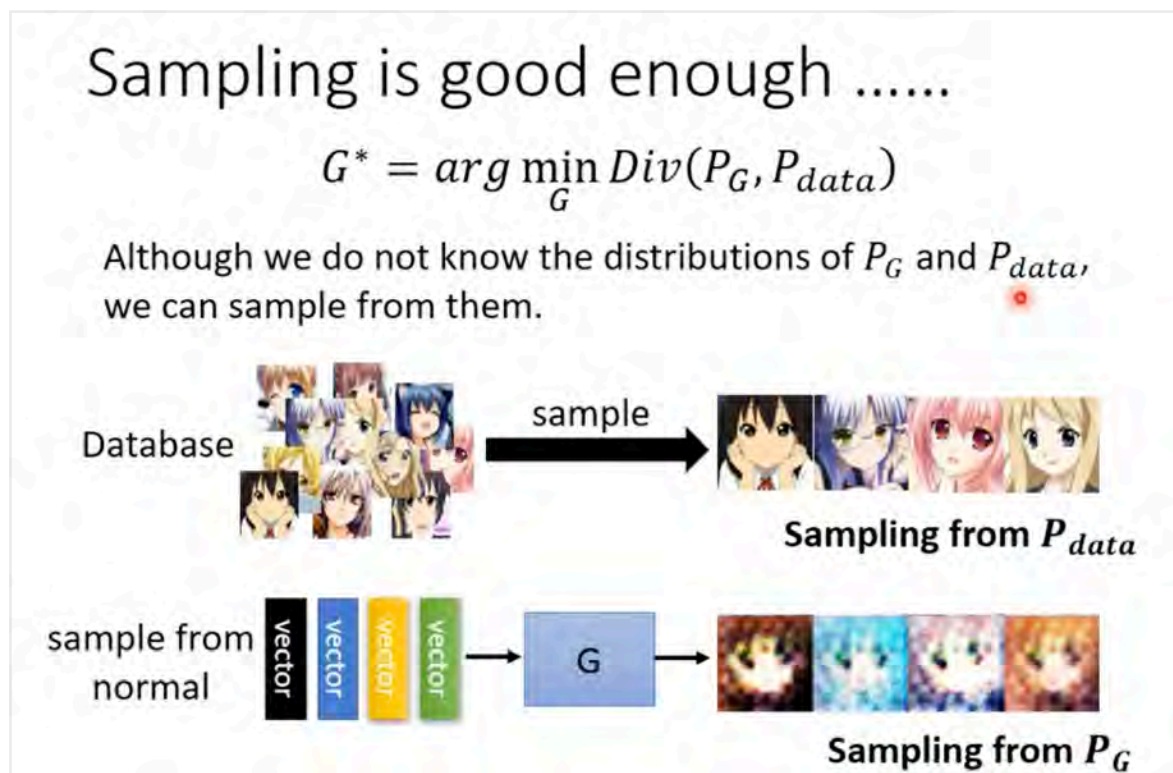


Theory behind GAN

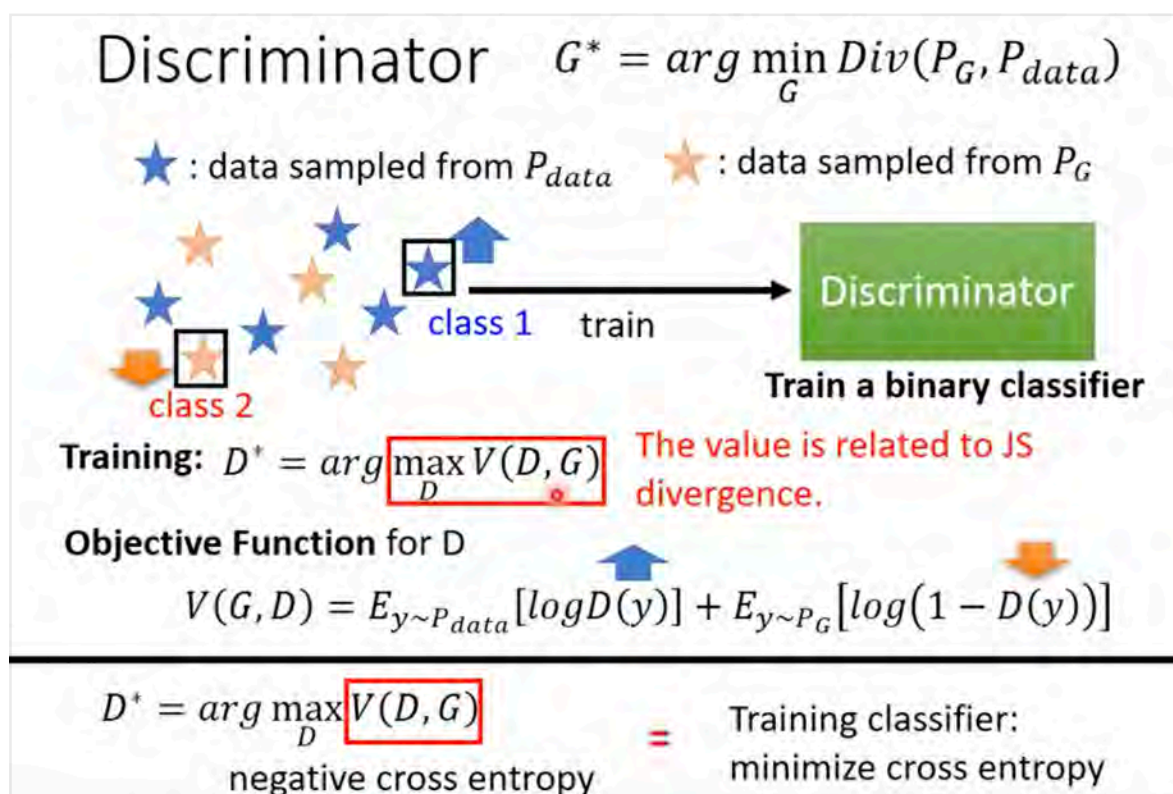


优化目标：minimize Divergence between P_G and P_{data} （用 generator 生成的样本的分布要和 data 本来的分布（用 discriminator 来判定）

Divergence:

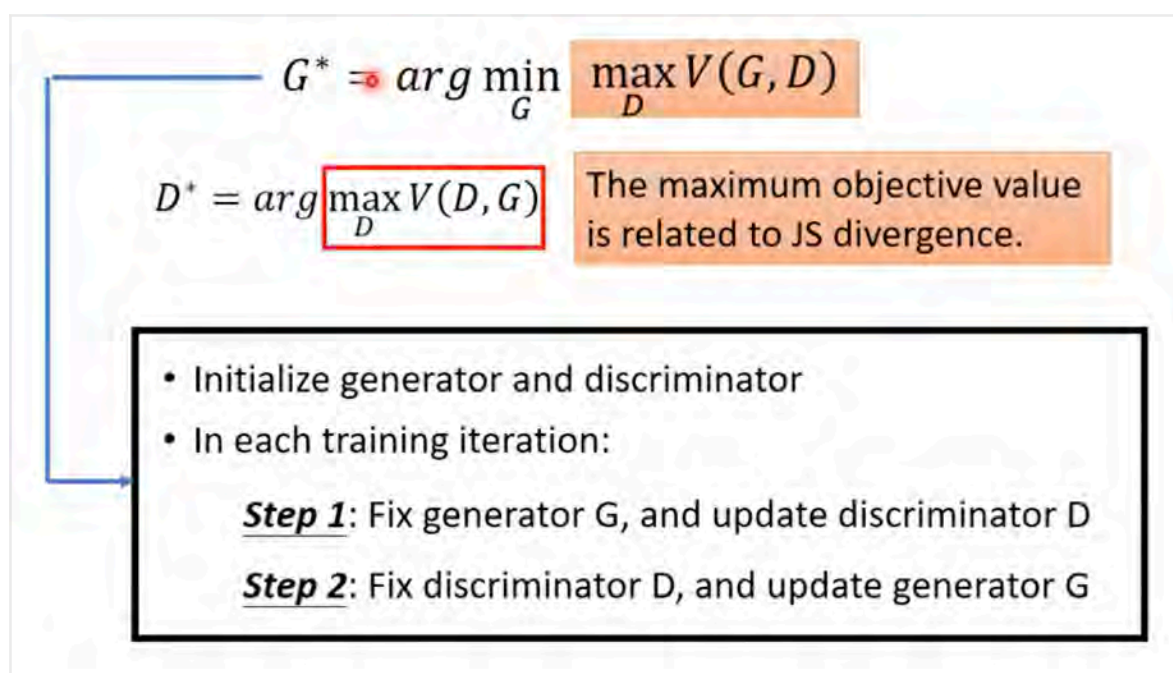


训练 discriminator:



直观理解就是当 GAN 训练的比较好的时候，有 $\max V(D, G)$ 和 Max Objective Function 相当于 $\min \text{Div}$

最终的优化目标：



Training skills:

在高维空间下目标 P_{data} 可能和 P_G 的重叠量非常少，而且我们采用采样，可能忽略更多的相近性

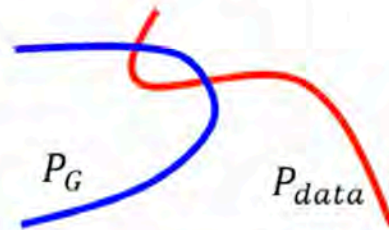
JS divergence is not suitable

- In most cases, P_G and P_{data} are not overlapped.

- 1. The nature of data

Both P_{data} and P_G are low-dim manifold in high-dim space.

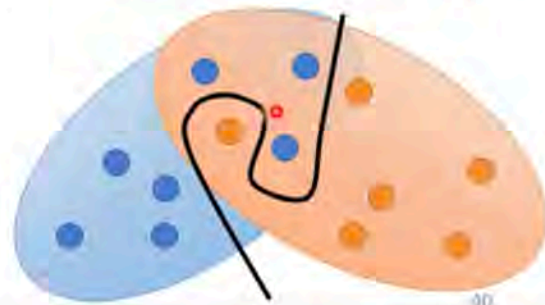
The overlap can be ignored.



- 2. Sampling

Even though P_{data} and P_G have overlap.

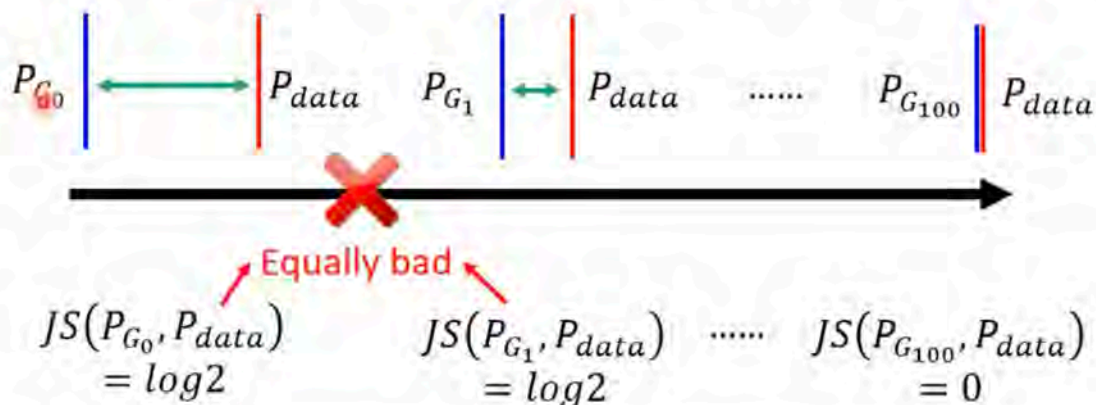
If you do not have enough sampling



JS divergence: binary 太简单了, 很容易就过拟合

What is the problem of JS divergence?

JS divergence is always $\log 2$ if two distributions do not overlap.

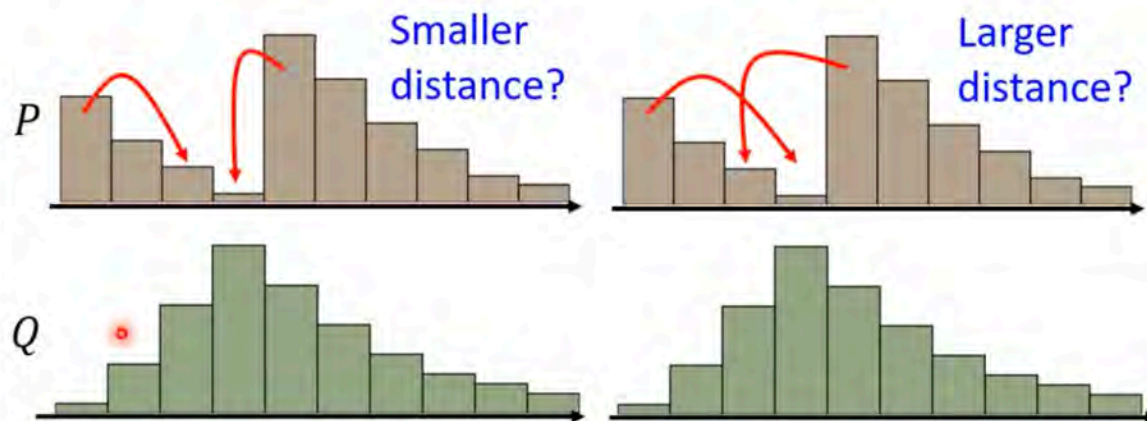


Intuition: If two distributions do not overlap, binary classifier achieves 100% accuracy.

The accuracy (or loss) means nothing during GAN training.

wasserstein distance

Wasserstein distance

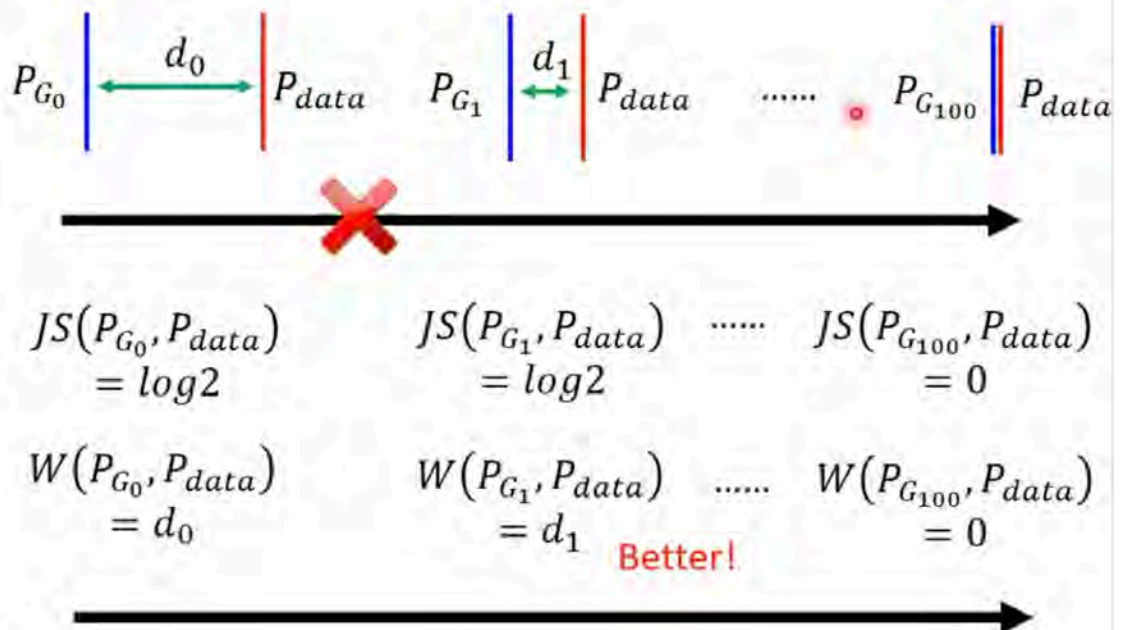


There are many possible "moving plans".

Using the "moving plan" with the smallest average distance to define the Wasserstein distance.

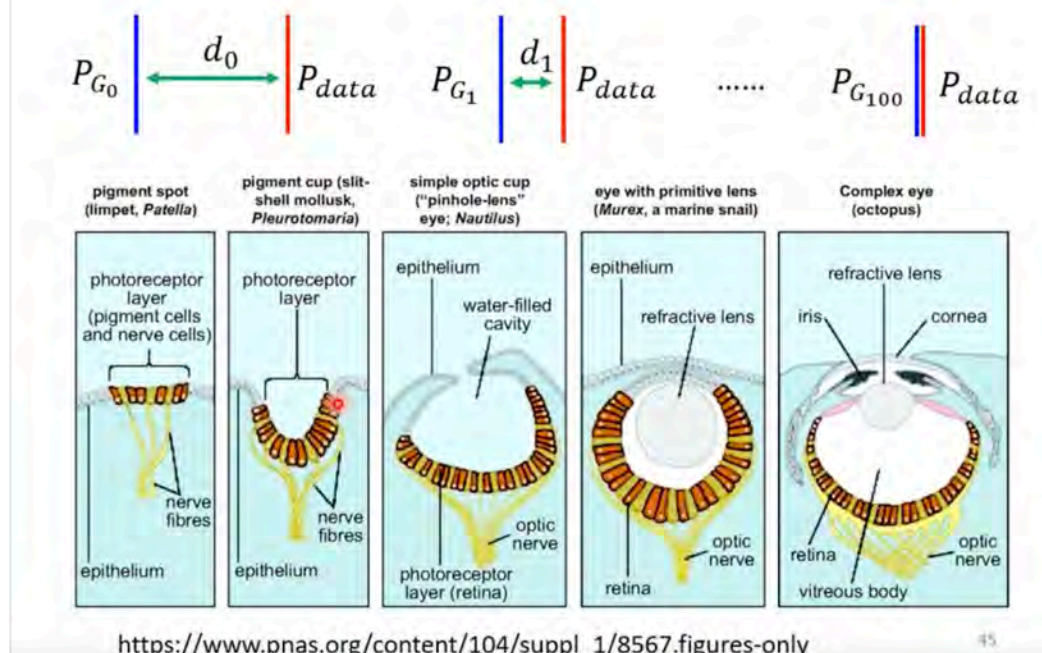
Why wasserstein distance better

What is the problem of JS divergence?



Eg. 人眼

What is the problem of JS divergence?



怎么 maximize W Distance:

WGAN

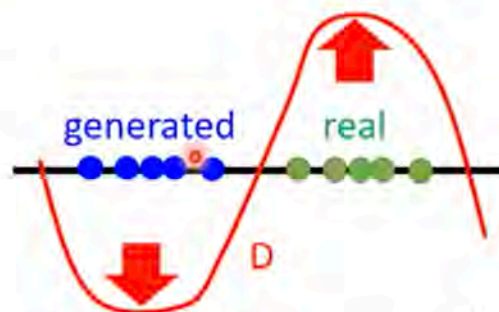
Evaluate Wasserstein distance between P_{data} and P_G

$$\max_{D \in 1-Lipschitz} \{E_{x \sim P_{data}}[D(x)] - E_{x \sim P_G}[D(x)]\}$$

D has to be smooth enough.

Without the constraint, the training of D will not converge.

Keeping the D smooth forces $D(x)$ become ∞ and $-\infty$

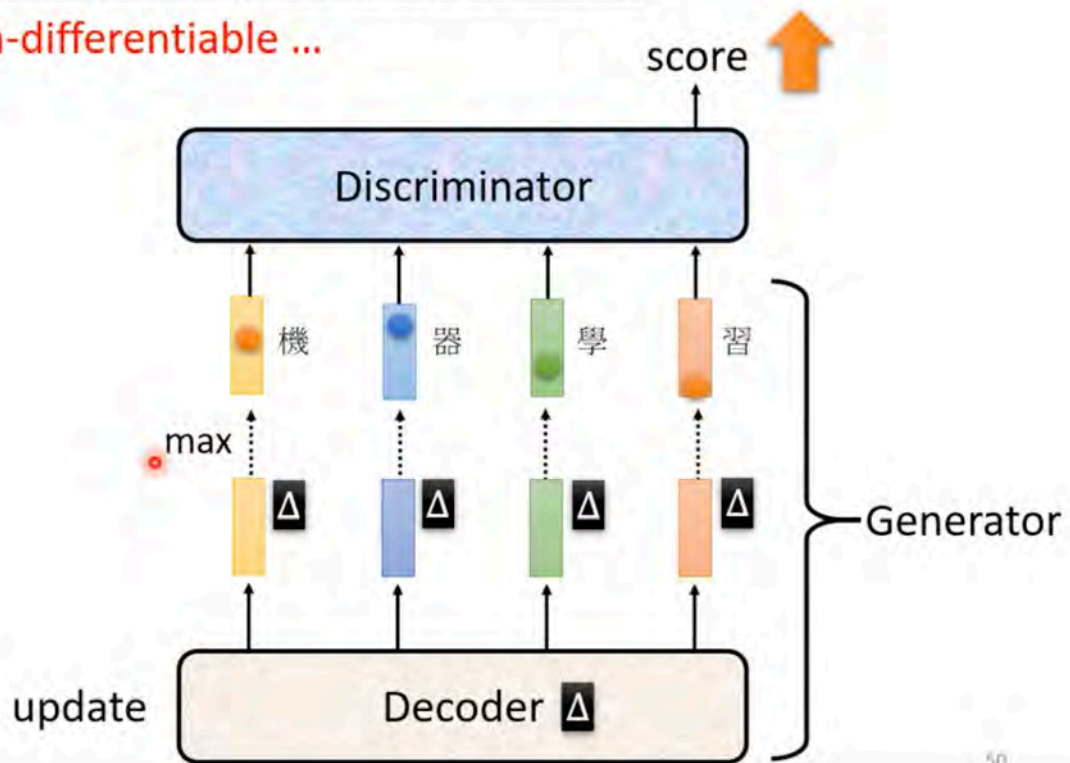


GAN 训练 2

Seq2Seq 模型: max 可能还是不变, 因为实质上这是一个 argmax (不可导), 选择最大那个作为输出, 而 CNN 里面的 Max 是用来做池化, 并没有依据 max 函数来选择输出 (池化层不 pick, pick 在 fc 后)

GAN for Sequence Generation

Non-differentiable ...



50

GAN evaluation

1. 后接一个classification, 看分类结果咋样

problem: mode collapse

Diversity - Mode Collapse



59

Mode dropping

Diversity - Mode Dropping

★ : real data

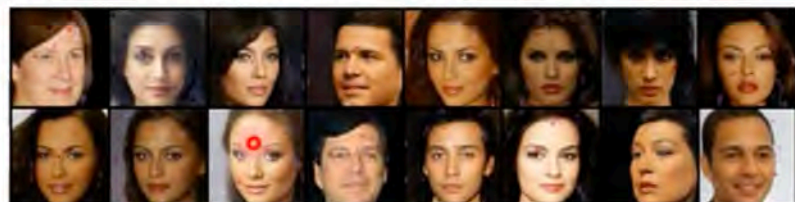
★ : generated data



Generator
at iteration t



Generator
at iteration t+1



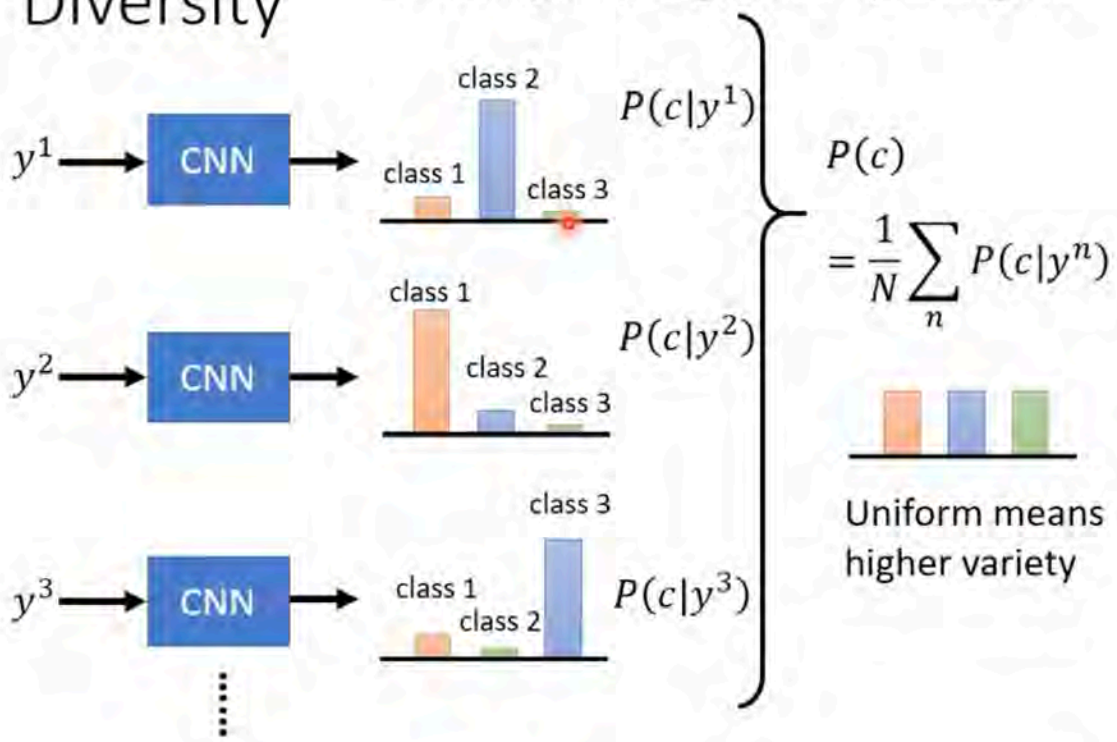
(BEGAN on CelebA)

平均分布的样本拥有更高的 diversity

Diversity

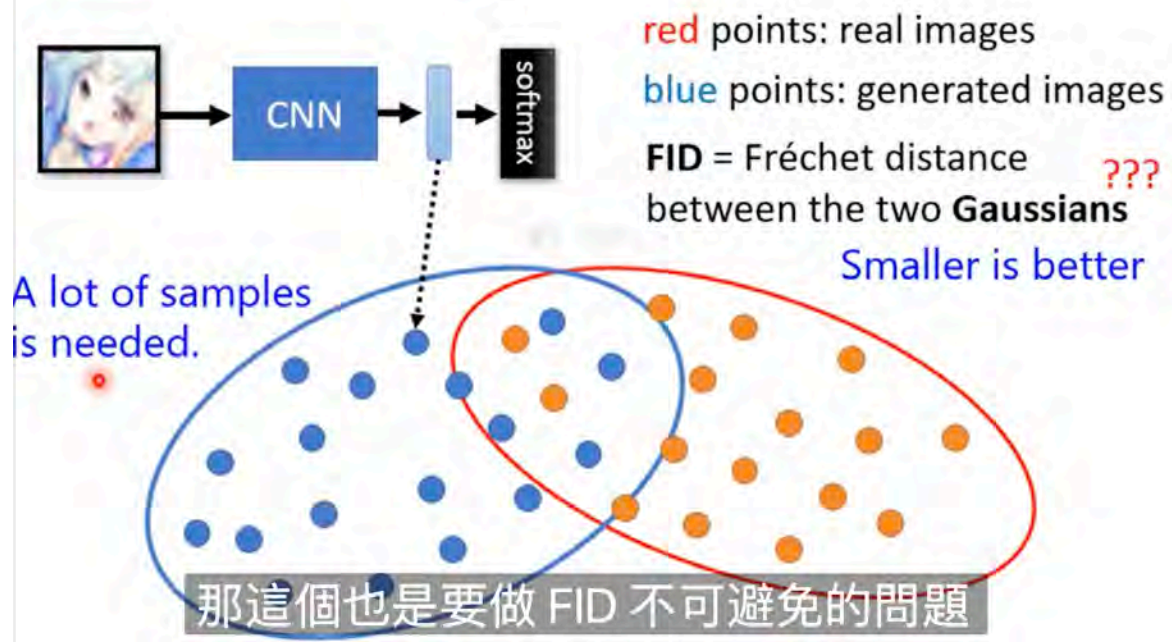
Inception Score (IS):

Good quality, large diversity → Large IS



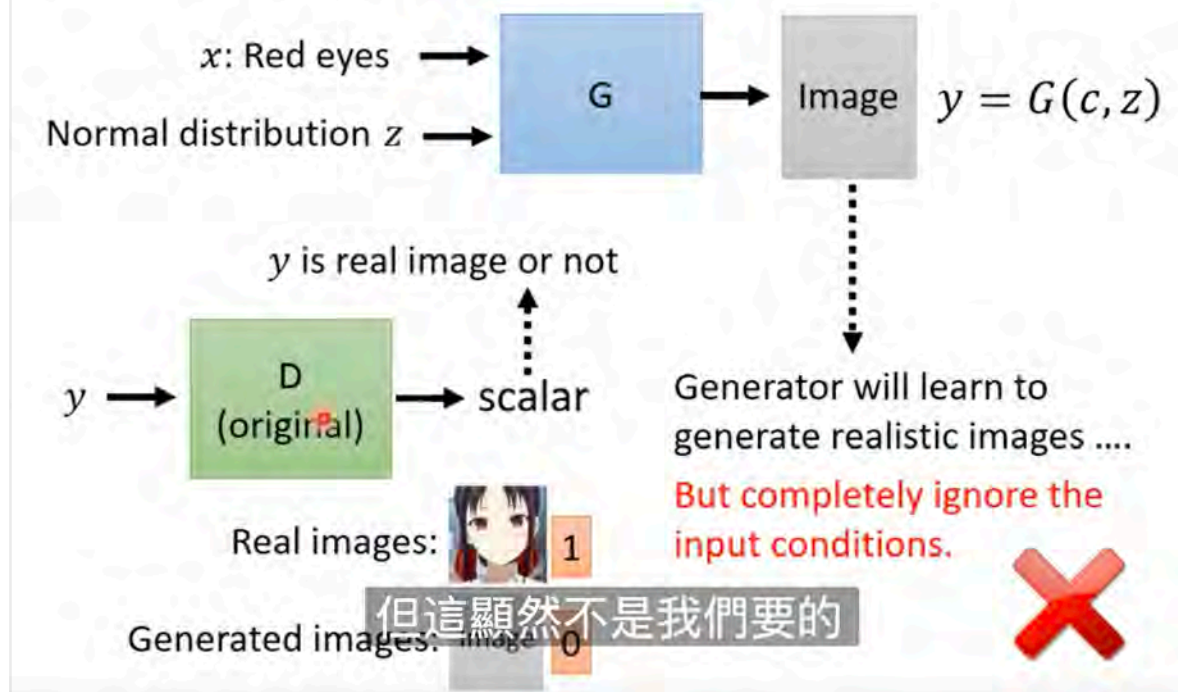
FID 可以用来计算一个复杂图像的 diversity

Fréchet Inception Distance (FID)



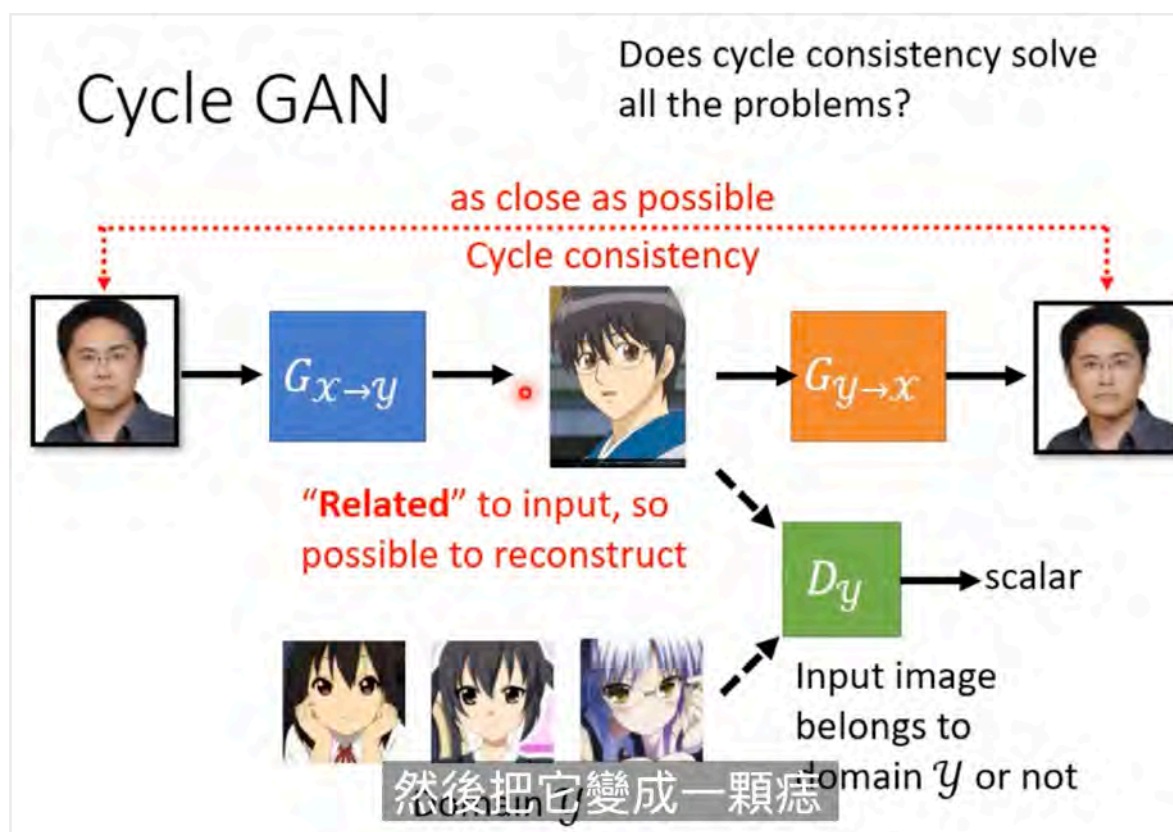
condition GAN (可以实现多模态的数据)

Conditional GAN



Unsupervised Learning —GAN approach

Semi-supervised Learning 可以用 pseudo label 来 match unpaired data, 但也有些问题是完全没有 train set 的



4. NLP

分词包 (jieba)

算法

- 基于前缀词典实现高效的词图扫描，生成句子中汉字所有可能成词情况所构成的有向无环图 (DAG)
- 采用了动态规划查找最大概率路径，找出基于词频的最大切分组合
- 对于未登录词，采用了基于汉字成词能力的 HMM 模型，使用了 Viterbi 算法

命名实体识别 (Named Entity Recognition)，简称 **NER**，是指识别文本中具有特定意义的实体，主要包括人名、地名、机构名、专有名词等，以及时间、数量、货币、比例数值等文字。

Hidden Markov Model, HMM

NER本质上可以看成是一种序列标注问题（预测每个字的 BIOES 标记），在使用 HMM 解决 NER 这种序列标注问题的时候，我们所能观测到的是字组成的序列（观测序列），观测不到的是每个字对应的标注（状态序列）。

文本情感分析主要有三大任务，即文本情感特征提取、文本情感特征分类以及文本情感特征检索与归纳。而关于文本情感分析的方法主要分为两类：

基于情感词典的方法

人工构建情感词典

自动构建情感词典

基于机器学习的方法

朴素贝叶斯

最大熵

SVM分类器 (binary)

包: TextBlob

可以做分句

blob.sentences

blob.sentences[0].sentiment

```
In [3]: blob.sentences
```

```
Out[3]: [Sentence("I am happy today."), Sentence("I feel sad today.")]
```

```
In [4]: blob.sentences[0].sentiment
```

```
Out[4]: Sentiment(polarity=0.8, subjectivity=1.0)
```

情感极性 0.8, 主观性 1.0。说明一下, 情感极性的变化范围是 $[-1, 1]$, -1代表完全负面, 1代表完全正面。

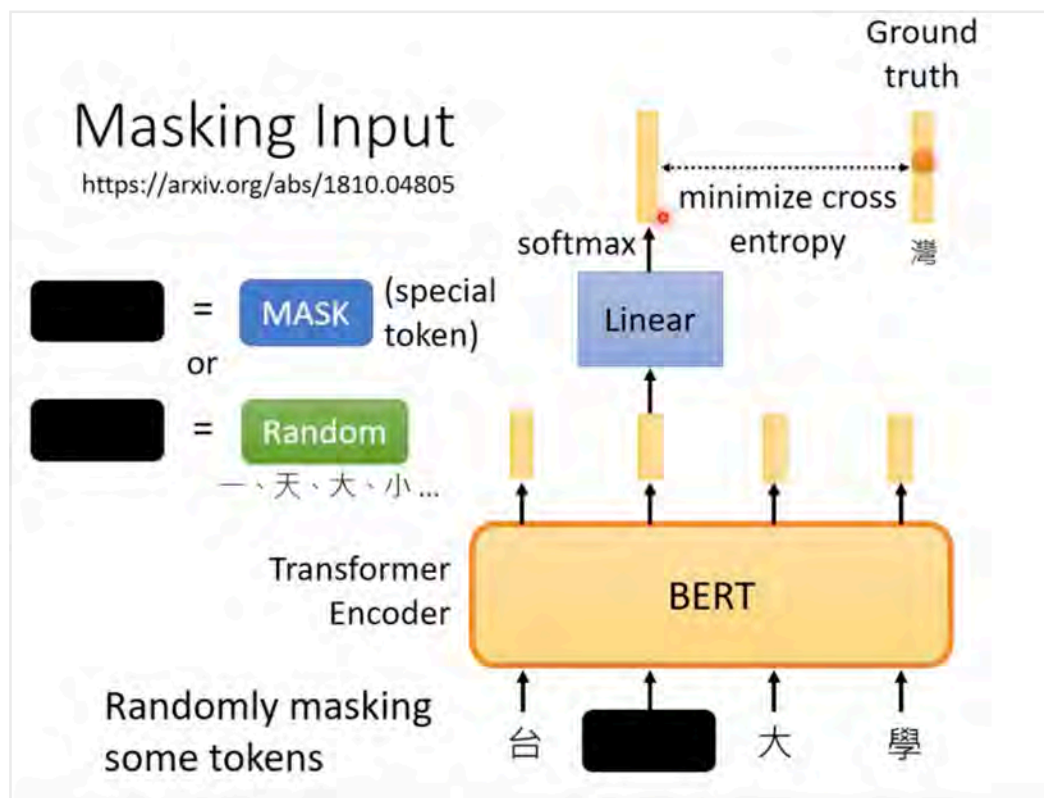
SnowNLP和blob差不多, 可以用来处理中文文本

Bert

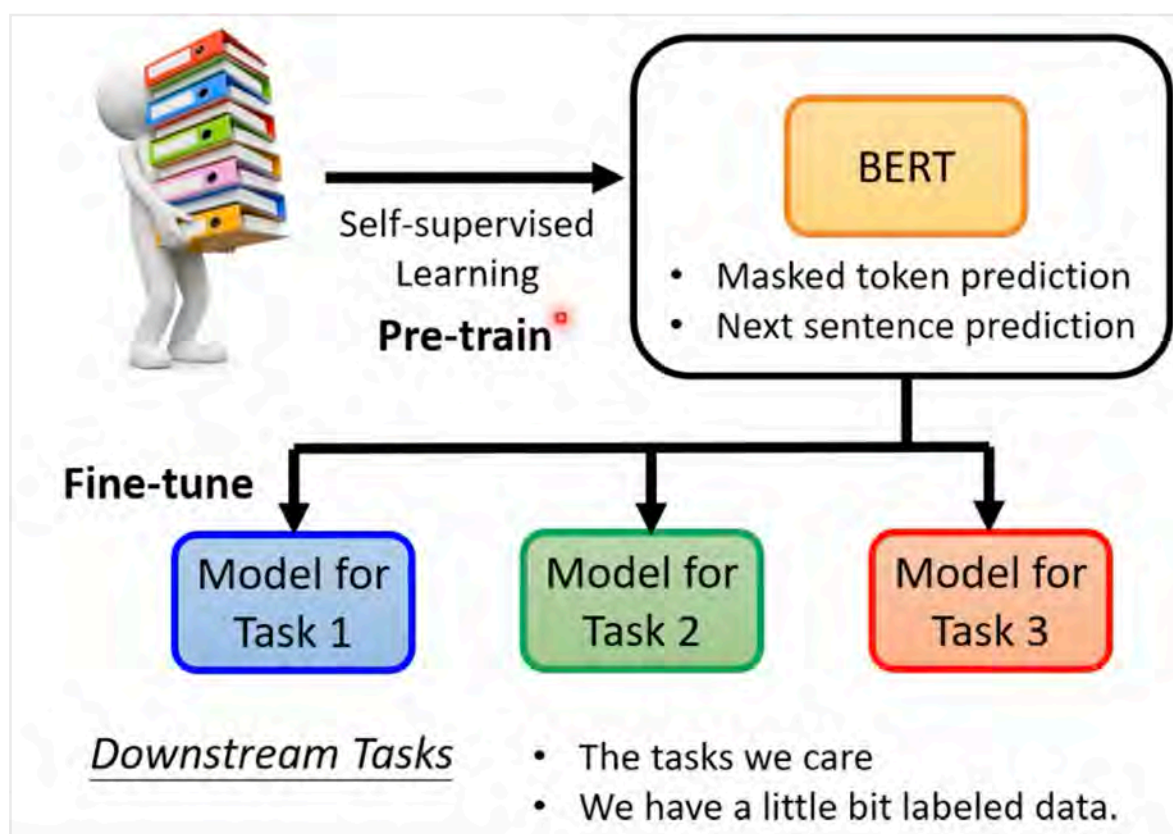
NLP中的 transfer learning

过去: 使用预训练的模型来抽取词 (word2vec), 不更新预训练的模型, 忽略了底层的时序特征

masking input:



BERT 是 pre trained 的, 只需要 fine tuning 就可以迁移到别的任务上去



完成 pre-train 之后：

